NVMesh Diag - usage and installation

Description

NVMesh Diag is a tool which can be used to capture hardware and system configuration details required to perform health checks, check for crucial NVMesh software dependencies, verify best practices and to gather vital information for installation and deployment planning.

Synopsis

nvmesh_diag.py [-sv]

Options

-v

This option enables the verbose and detailed output on the screen and command line. The default setting is disabled. In case the tool is left to the defaults, the information can be looked up later on in the *.nvmesh_diag_putput.txt file which will be created by the tool in the same folder as its being executed in.

-s

Enables the interactive tuning of the system and adjustment of system settings according to the NVMesh best practices. The current version will allow you to set the required:
- SELinux or AppArmor settings
- Firewall settings
- Tuned settings
- IRQ balancer settings
And will install the Mellanox inbox drivers in case OFED is not installed and not required.

Logging

The tool logs the activities, warnings, and errors the in the nvmesh_diag.log file, which will be created by the tool in the same folder as its being executed in.

Operating System Support

CentOS and RHEL 7.3 and 7.4
SuSE Enterprise Linux Server 12, SP3

Installation

One easy and quick way to install and get ready for use is using git and clone the project directly on the server.

The link to the git repository is here: https://github.com/Excelero/nvmesh_diag

git clone

```
[root@hostname ~]# git clone https://github.com/Excelero/nvmesh_diag
Cloning into 'nvmesh_diag'...
remote: Counting objects: 19, done.
remote: Compressing objects: 100% (18/18), done.
remote: Total 19 (delta 4), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (19/19), done.
```

The next step is to change into the nvmesh_diag directory and make the nvmesh_diag.py executable.

chmod

```
[root@hostname ~]# cd nvmesh_diag
[root@hostname nvmesh_diag]# chmod +x nvmesh_diag.py
```

To run the tool, just execute the script. Please see sample output as below.Tuned profile settings are OK

./nvmesh_diag

```
[root@hostname nvmesh_diag]# ./nvmesh_diag.py
Collecting Host Name Information:
Done.

Collecting Hardware Vendor and System Information:
Done.

Collecting Operating System Information:
Red Hat Enterprise Linux Server 7.3 3.10.0-514.26.2.el7.x86_64
Please verify this information with the latest support matrix!
Done.

Collecting and Verifying SELinux Information:
Disabled - OK

Collecting and Verifying Firewall Information:
Disabled - OK

Collecting and Verifying CPU Information:
2 Physical CPU
Intel(R) Xeon(R) CPU E5-2690 v4 @ 2.60GHz
CPU frequency settings OK.

Collecting and Verifying System Tuning Information:
```

```
Tuned profile settings are OK
Checking the IRQ balancer...
The IRQ balancer is running - OK

Collecting Memory Information:
Done.

Collecting High Level Block Device Information:
Done.

Collecting NVMe Storage Device Information:
Found that NVMesh software components are installed already. Checking
the NVMesh services now...
nvmeshclient status [  OK  ]

Collecting And Verifying Mellanox Driver Information:
OFED installed - OK
Version: 4.2-1.2.0.0
Please verify this information with the latest support matrix!

Collecting and Verifying R-NIC information:
Checking HCA at PCIe address: 03:00.0
 Vendor/OEM information: Subsystem: Mellanox Technologies
 HCA Type:  CX515A - ConnectX-5 QSFP28
 Firmware level: 16.21.2010
 HCA PCIe speed settings OK. Running at 8GT/s
 HCA PCIe width settings OK. Running at x16

Checking HCA at PCIe address: 85:00.0
 Vendor/OEM information: Subsystem: Mellanox Technologies
 Firmware level: 2.42.5000
 HCA PCIe speed settings OK. Running at 8GT/s
 HCA PCIe width settings OK. Running at x8


Collecting And Verifying RDMA Specific Information:
mlx5_0 guid OK
mlx5_0 ready and configured for RDDA
Done.

Collecting Infiniband Specific Information:
Done.
```

```
Collecting IP Address Information:
Done.
```